UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/021,606	12/10/2001	Clifford A. Whitehill	01-676	7224

7590 09/23/2004

EXAMINER
INGBERG, TODD D

ART UNIT	PAPER NUMBER
2124	

Peter Scott
LSI Logic Corporation
Intellectual Property Law Department
1551 McCarthy Boulevard, M/S D-106
Milpitas, CA 95035

DATE MAILED: 09/23/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	10/021,606	WHITEHILL, CLIFFORD A.	
	Examiner	Art Unit	
	Todd Ingberg	2124	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *10 December 2001*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-15* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-15* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *10 December 2001* is/are: a)☐ accepted or b)☒ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

Claims 1 – 15 have been examined.

Drawings

1. Drawings filed December 10, 2001 are deemed informal. New formal drawings are required with response to this office action.

Claim Rejections - 35 USC § 101

2. 35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

3. Claims 1 – 15 are rejected under 35 USC §101 for not being concrete and tangible. The rejection can easily be overcome. The Examiner has provided one such way to overcome the rejection.

Claim 1
A method **executing on a computer and stored on a computer readable medium** for implementing bit-fields for a machine word comprising: defining a variable corresponding to the machine word; defining at least one bit-field within the variable, the bit-field having a first bit and a second bit; for each bit-field, determining a first bit number defining a position of the first bit of the bit-field within the variable; for each bit-field, determining a second bit number defining a position of the second bit of the bit-field within the variable; and
for each bit-field, concatenating the first bit number and second bit number together to form a bit-field definition number defining a location of the bit-field within the variable.

Claim 9
A method **executing on a computer and stored on a computer readable medium** of handling a bit-field for a machine word comprising: defining a machine word having a plurality of bits including a bit-field located from a first one of the bits to a second one of the bits in the machine word; forming a bit-field definition number having a first portion and a
second portion, the first portion formed from a location of the first bit in the machine word, the second portion formed from a location of the second bit in the machine word, the bit-field definition number defining a location and range of the bit-field in the machine word; and associating the bit-field definition number with the bit-field in the machine word.

Claim Rejections - 35 USC § 102

4. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

5. Claims 1 , 3 and 9 - 10 are rejected under 35 U.S.C. 102(b) as being anticipated by Admitted Prior Art of Record.

Claim 1
A method for implementing bit-fields for a machine word comprising: defining a variable corresponding to the machine word; defining at least one bit-field within the variable, the bit-field having a first bit and a second bit; for each bit-field, determining a first bit number defining a position of the first bit of the bit-field within the variable; for each bit-field, determining a second bit number defining a position of the second bit of the bit-field within the variable; and for each bit-field, concatenating the first bit number and second bit number together to form a bit-field definition number defining a location of the bit-field within the variable.
Examiner's Response
Prior art figures 1 – 3 disclose how to perform the limitations of claim 1 in the programming languages C and C++. **Note**, the Applicant has not claimed the definition of a programming construct that define macro(s) to perform the invention as the actual invention as disclosed in the instant invention.

Claim 3
A method as defined in claim 1 further comprising: for each bit-field, determining the first bit number by determining a left bit of the bit-field; and for each bit-field, determining the second bit number by determining a right bit of the bit-field.
Examiner's Response
As per claim 1 and the prior art figure 6 which shows the bit wise operators for shifting in the programming languages C and C++.

Claim 9
A method of handling a bit-field for a machine word comprising: defining a machine word having a plurality of bits including a bit-field located from a first one of the bits to a second one of the bits in the machine word; forming a bit-field definition number having a first portion and a second portion, the first portion formed from a location of the first bit in the machine word, the second portion formed from a location of the second bit in the machine word, the bit-field definition number defining a location and range of the bit-field in the machine word; and associating the bit-field definition number with the bit-field in the machine word.

Examiner's Response
Prior art figures 1 – 3 disclose how to perform the limitations of claim 1 in the programming languages C and C++. **Note**, the Applicant has not claimed the definition of a programming construct that define macro(s) to perform the invention as the actual invention as disclosed in the instant invention.

Claim 10
A method as defined in claim 9 further comprising: forming the bit field definition number by concatenating a number of the first bit and a number of they second bit together, the first portion and the second portion representing the numbers of the first and second bits, respectively;
Examiner's Response
Prior art figures 1 – 3 disclose how to perform the limitations of claim 1 in the programming languages C and C++.

Claim Rejections - 35 USC § 103

6. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

7. Claims 2, 4 – 7 and 11 - 15 rejected under 35 U.S.C. 103(a) as being unpatentable over Admitted Prior Art of the programming languages of C and C++ (as illustrated in Figures 1 – 6) for bit wise operations in view of the IDE of Visual C++ as documented by Ivor Horton, Beginning Visual C++ 5, March 19, 1997.

Claim 2
A method as defined in claim 1, wherein the variable includes a structure, further comprising: defining the variable in a structure declaration; and for each bit-field (as per clam 1), defining an alias for the bit-field definition number within the structure declaration.
Examiner's Response
Prior Art figures do not show how to define alias structures. It is a supported operation in the programming languages of both C and C++ to be able to define alias structures. Visual C++ teaches the use of pointers to access via an alias to variables within a struct. (VC++, page 240). The use of pointers with a struct is grossly old and well known and well with in the abilities of

one of very ordinary skill at the time of invention. Therefor it would have been obvious to modify the Admitted Prior art of record to use pointers because pointers reduce the amount of typing for a programmer.

Claim 4

A method as defined in claim 1 further comprising: selecting one of the at least one bit-field from which data is to be read; extracting the first and second bit numbers from the bit-field definition number for the selected bit-field; generating a mask of the variable from the extracted first and second bit numbers to isolate the selected bit-field; using the mask to zero-out all bits of the variable except for within the selected bit-field; and using one of the first and second bit numbers to shift the selected bit-field to form the data to be read.

Examiner's Response

The Admitted prior art teaches extracting of bits is part of the admitted prior art figures as shown in figure 6. But does not show The limitations of zeroing out bits is met by the initialization of data performed as taught by Visual C++ on page 235 and the building of a mask (Window) as taught on page 530 of Visual C++. Building of masks in Visual C++ and initializing variables is grossly old and well known. Therefore, it would have been obvious to one of ordinary skill in the art to combine the Admitted Prior art with Visual C++ because visual programming is more intuitive.

Claim 5

A method as defined in claim 1 further comprising: selecting one of the at least one bit-field to which data is to be written; providing a write value to be written into the selected bit-field; extracting the first: and second bit numbers from the bit-field definition number for the selected bit-field; generating a mask of the variable from the extracted first and second bit numbers to isolate the selected bit-field; using the mask to zero-out all bits of the variable only within the selected bit-field; using one of the first and second bit numbers to shift the write value to the location of the selected bit-field; and inserting the write value into the selected bit-field.

Examiner's Response

As per claim 4.

Claim 6

A method as defined in claim 1 further comprising: selecting one of the at least one bit-field for which a mask is to be formed; extracting the first and second bit numbers from the bit-field definition number for the selected bit-field; subtracting the second bit number from the first bit number to form a first shift value; providing a mask value having one bit of a first logical value and remaining bits of a second logical value, the first logical value bit being an end bit of the initial value; shifting the first logical value bit from the end bit of the mask value by the first shift value, all bits between the first logical value bit and the end bit inclusive being first logical value bits, and the remaining bits having the second logical value; inverting all of the bits of the mask value, the first logical value bits becoming the second logical value bits, and the remaining bits having the first logical value; and forming the mask for the selected bit-field by using the second bit number as a second shift value! and shifting the second logical value bits from the end bit of

the mask value by the second shift value, all bits between the second logical value bits and the end bit having the first logical value.

Examiner's Response

As per claim 4.

Claim 7

A method as defined in claim 1 further comprising: selecting one of the at least one bit-field; extracting the first bit number from the bit-field definition number for the selected bit-field by dividing the bit-field definition number by a divisor and discarding any remainder from the division; and extracting the second bit number from the bit-field definition number for the selected bit-field by dividing the bit-field definition number by the divisor and discarding any non-remainder from the division.

Examiner's Response

The Admitted prior art teaches extracting of bits is part of the admitted prior art figures as shown in figure 6. But does not show Building of a mask (Window) as taught on page 530 of Visual C++. And Visual C++ teaches bitwise operators on pages 68 – 69 and the importance of maintaining the bits wanted .Building of bitwise operators and masks in Visual C++ is grossly old and well known. Therefore, it would have been obvious to one of ordinary skill in the art to combine the Admitted Prior art with Visual C++ because visual programming is more intuitive and the use of bitwise operators ".. should only, use them in the way if you are sure you are not going to lose bits that you can ill afford to be without." (Visual C++, page 69) .

Claim 8

A method as defined in claim 7 wherein the divisor is 100.

Examiner's Response

Visual C++ teaches the use of bitwise operators where the bits represent values. In the Example the number 24 is divided by 10 where the number 4 is discarded. One of ordinary skill in the art would know to shift a value of 240 an additional bit to make the divisor 100 to produce the remainder of 2. Therefore, it would have been obvious to one of ordinary skill in the art to combine the Admitted Prior art with Visual C++ because the use of bitwise operators ".. should only, use them in the way if you are sure you are not going to lose bits that you can ill afford to be without." (Visual C++, page 69) .

Claim 11

A method as defined in claim 10 further comprising: retrieving the number of the first bit from the first portion of the bit-field definition number by dividing the bit-field definition number by a divisor and discarding any remainder; and retrieving the number of the second bit from the second portion of the bit-field definition number by dividing the bit-field definition number by the divisor and discarding any non-remainder.

Examiner's Response

As per claim 7.

Claim 12

A method as defined in claim 9 further comprising: extracting the first and second portions from the bit-field definition number; and generating a mask of the bit-field in the machine word from the extracted first and second portions.

Examiner's Response

The Admitted prior art teaches extracting of bits is part of the admitted prior art figures as shown in figure 6. But does not show Building of a mask (Window) as taught on page 530 of Visual C++. Building of masks in Visual C++ is grossly old and well known. Therefore, it would have been obvious to one of ordinary skill in the art to combine the Admitted Prior art with Visual C++ because visual programming is more intuitive.

Claim 13

A method as defined in claim 12 further comprising: generating the mask of the bit-field in the machine word by: calculating a difference between the extracted first and second portions of the bit-field definition number; shifting a predetermined value by the calculated difference; inverting the shifted predetermined value; and shifting the inverted shifted predetermined value by the second portion of the bit-field definition number.

Examiner's Response

The Admitted prior art teaches extracting of bits is part of the admitted prior art figures as shown in figure 6. But does not show Building of a mask (Window) as taught on page 530 of Visual C++. Building of masks in Visual C++ is grossly old and well known. Therefore, it would have been obvious to one of ordinary skill in the art to combine the Admitted Prior art with Visual C++ because visual programming is more intuitive.

Claim 14

A method as defined in claim 12 further comprising: reading a current value of the bit-field by: ANDing the machine word with the generated mask to form an ANDed result having all the bits of the machine word outside of the bit-field zeroed-out; and shifting the ANDed result by the second portion of the bit-field definition number.

Examiner's Response

The Admitted prior art teaches extracting of bits is part of the admitted prior art figures as shown in figure 6. But does not show The limitations of zeroing out bits is met by the initialization of data performed as taught by Visual C++ on page 235 and the building of a mask (Window) as taught on page 530 of Visual C++ and use of well known bit wise operator as taught on Visual C++ , pages 64 – 69. Building of masks in Visual C++ and initializing variables and bit wise operators are grossly old and well known. Therefore, it would have been obvious to one of ordinary skill in the art to combine the Admitted Prior art with Visual C++ because visual programming is more intuitive.

Claim 15

A method as defined in claim 12 further comprising: writing a write value to the bit-field by: inverting the generated mask; logically ANDing the machine word with the inverted generated mask to form an ANDed result having the bits of the machine word only within the bit-field zeroed-out; shifting the write value by the second portion of the bit-field definition number; and logically ORing the shifted write value with the ANDed result.

Examiner's Response

The Admitted prior art teaches extracting of bits is part of the admitted prior art figures as shown in figure 6. But does not show The limitations of zeroing out bits is met by the initialization of data performed as taught by Visual C++ on page 235 and the building of a mask (Window) as taught on page 530 of Visual C++ and use of well known bit wise operator as taught on Visual C++ , pages 64 – 69. Building of masks in Visual C++ and initializing variables and bit wise operators are grossly old and well known. Therefore, it would have been obvious to one of ordinary skill in the art to combine the Admitted Prior art with Visual C++ because visual programming is more intuitive.

Conclusion

8. The prior art rejection can easily be overcome by claiming of macro definition/defining in a mask for defining a bit-field operations with the limitations of the independent claims present. The current claim language reads on defining bit-filed definitions which in include defining such operations in an Integrated Development Environment (IDE) such as Visual C++ using the prior art shown in figures 1- 6.

Correspondence Information

Any inquiry concerning this communication or earlier communications from the examiner should be directed to **Todd Ingberg** whose telephone number is (703) 305-9775.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, **Kakali Chaki** can be reached on (703) 305-9662. Please, note that as of August 4, 2003 the **FAX number** changed for the organization where this application or proceeding is assigned is **(703) 872-9306**.

Also, be advised the United States Patent Office **new address** is

Post Office Box 1450

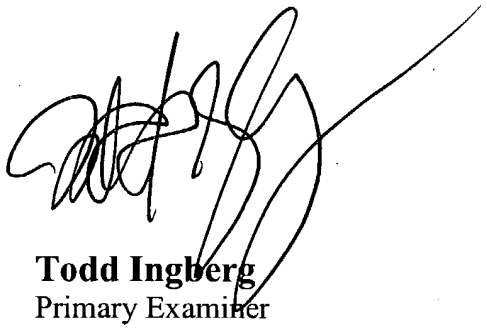
Alexandria, Virginia 22313-1450

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (703) 305-9700.

Special Notice

Please, Note the Examiner's telephone number will change in October when the Art Unit moves to the new location. The Examiner's new telephone number will be as follows:

(571) 272-3723



Todd Ingberg
Primary Examiner
Art Unit 2124
September 19, 2004